The Interplay between Model Coverage and Code Coverage
Mirko Conrad, André Baresel, Sadegh Sadeghipour, Joachim Wegener

Mirko Conrad holds a degree in Computer Science from the Technical University Berlin, Germany. He has been a research scientist in the Software Technology Lab of DaimlerChrysler Research & Technology since 1995. His industrial experience includes model-based development and testing of embedded automotive software. He was involved in the development of the model-based testing environment MTest. He is a member of the Special Interest Group for Testing, Analysis and Verification of Software in the German Computer Society (GI).

André Baresel has finished his study at the Humboldt University of Berlin, Germany with a degree in Computer Science. Since 2000 he works as research scientist in the Software Technology Research Department of DaimlerChrysler AG. His work is currently focused on the development of a tool environment for automatic test data generation with evolutionary algorithms.

Sadegh Sadeghipour holds a PhD degree in Computer Science from the Technical University Berlin. He joined the Software Technology Lab of DaimlerChrysler Research & Technology as PhD student in 1995. Between 1998 and 2000 he worked there as research scientist. At present, he conducts the software consulting company ITPower in Berlin, which was co-founded by him in 2000. Sadegh has presented several articles on specification-based testing in international conferences and journals since 1996.

Joachim Wegener holds a PhD in Computer Science from Humboldt University Berlin, Germany. He is manager of Adaptive Technologies in Software Engineering at DaimlerChrysler, Research and Technology. He was involved in the development of the classification-tree editor CTE, the test system TESSY and the automatic structural test system ET. He is working on systematic and evolutionary software methods for the verification of embedded systems. He is a member of SAE International, the Seminal network and the German Computer Society Interest Group on Testing, Analysis and verification.
The Interplay between Model Coverage and Code Coverage

A. Barresi, M. Conrad, J. Wegener
S. Sadeghipour
DaimlerChrysler
Electrical Engineering

Model-based Development & Testing
- Model vs. Code Coverage
- Experimental Studies
- Lessons learned

Model-based Development
Modelling Tools and Techniques
- execute models (e.g. Simulink / Stateflow or TargetLink)
- block diagrams and extended state machines
- managing complexity by means of hierarchical decomposition
- automatic generation of C code
- commercial modelling and simulation environments (e.g. Matlab / Simulink / Stateflow) and code generators (e.g. TargetLink)

Model Coverage vs. Code Coverage - Example

Model Coverage Metrics - Decision Coverage on Model Level (M_D1)
Pathways Through a Switch Block and Test Goals

Model-based Development
Paradigm Shift in Embedded Control Software Development
- An emerging strategy to reduce cost and development time is to replace the traditional “build, test, and fix” process with model-based product development methodologies.

traditional s/w development
- model-based development

integrated deployment of executable models for:
- specification
- design
- implementation and
- testing
Code Coverage Metrics - Branch Coverage on Code Level (C,C1)
Control Flow Graph and Test Goals

- Test goal: test input specification
  1. path A: control = threshold
  2. path B: control = threshold

Experimental Studies - Questions

- Which possible applications exist for model coverage measurements within the context of model-based development?
- Which correlation exists between structural model and code coverage criteria?
- How can model coverage analyses and test vectors be used to improve model-based testing?
- Can model coverage metrics supplement or, in part, replace the measurement of code coverage?

Experimental Studies - Procedure

- Test vector/test sequence generator + Test Code
- Stimuli/Stateflow model
- Model coverage analyzer/Model Coverage Test Tool/TargetLink
- Code coverage analyzer/TESSE
- Additional tests

Relationship between Model and Code Coverage - Example

- C code
- M_D1 model coverage report
- C_C1 code coverage report

Relationship between Model and Code Coverage - Correlation

- Calculated correlation:
  - between M_D1 and C_C1 is 0.98
  - between M_D1 and C_C0 is 0.97

- Under the given constraints, C_C0 and C_C1 code coverage can be approximated with M_D1 model coverage

Model Coverage Optimization

- Optimized model coverage
- Initial model coverage
Lessons Learned

- In addition to the common structural coverage criteria at code level, model-based development makes it also possible to determine structural test criteria at model level and integrates these into the control and evaluation of the test process.
- Model coverage measurements can be determined early in the development, i.e. before the program code is available. Relevant test activities could be carried out earlier resulting in early error detection and low-cost error correction.
- Comparable model and code coverage metrics exist, e.g. M_D1 model coverage seems to permit an estimation of the C_D1 code coverage to be expected.
- The exact correlation vector depending on the transformation algorithms used during the code generation -9 strong statistical correlation, but no 1:1 relationship
- In contrast to code coverage criteria which are defined tool-independent and are applicable to various imperative programming languages of the 3rd generation, model coverage metrics are tool-specific.
- By deploying automatic test vector / test sequence generation tools the structural coverage reached during black-box testing on both model and code level could be optimized.